

On Generating *-Sound Nets with Substitution

Jacek Sroka

University of Warsaw, Poland

Jan Hidders

Delft University of Technology, The Netherlands

Abstract

We present a method for hierarchically generating sound workflow nets by substitution of nets with multiple inputs and outputs. We show that our method is correct and generalizes the class of nets generated by other hierarchical approaches. We also identify a new notion of soundness and prove that it is preserved by substitutions. We claim that this notion is better suited than *-soundness for use with hierarchical methods, while showing examples that *-soundness in the general case is not preserved by substitution.

Keywords: Petri nets, workflow nets, soundness, substitution soundness, hierarchical nets, Petri-net refinement, AND-OR net

1. Introduction

Among all the different formalisms for modeling processes Petri nets [1] offer the distinct benefits of combining an easy-to-understand visual notation with a large body of practical and theoretical work on efficient and effectively reasoning over them. This has made them very popular for modeling of and reasoning over business processes and specifically business workflows. For describing the latter type of processes a specific class of Petri nets, called workflow nets, was introduced which feature attractive modeling and analytic properties such as easy to verify notions of correctness. They are defined as Petri nets with one input and one output place, representing the beginning and the end of the flow, that become strongly connected when we add a transition from the output place to the input place. The workflow starts with one token in the input place and while the workflow is running, it follows the usual firing rules of Petri nets. When the net has reached a state where it contains exactly one token, which is placed in the output place, then it is assumed to have completed correctly.

The correctness of any model, and therefore also workflow nets, ultimately depends on whether it correctly models the domain in question. However, certain

properties such as the absence of livelocks, deadlocks, and other anomalies are desirable and can be checked independently of the specific domain. Among these the *soundness* property of the workflow net is considered the most important. This notion was originally proposed by van der Aalst in [2] and since then several alternate notions of soundness have been proposed and studied. Informally speaking soundness means two things. First, that if we start with an initial token in the input place, then no matter how we proceed with the execution of the workflow, we can always end up in the final state with one token in the output place. Second, that every subtask can be potentially executed, i.e., there is at least one correct run of the workflow net in which this subtask is executed. An overview of the research on soundness of workflow nets with additional decidability results can be found in [3].

There are roughly two ways to guarantee that process modelers produce sound workflow nets. The first is to let them design the workflow as they like and then use the different existing algorithms to determine if the desired requirements are met, like in [4] or [5]. The second is to guide the design and allow only manipulations and combinations of nets that are guaranteed to produce sound nets [6]. In this paper we will follow the second approach and in particular focus on a structural approach where the net is constructed in a top-

Email addresses: sroka@mimuw.edu.pl (Jacek Sroka),
a.j.h.hidders@tudelft.nl (Jan Hidders)

down fashion. This means that the system is designed by first specifying a workflow net that provides a high-level description of the process by summarizing it at a high abstraction level in terms of high-level actions, and then refining this workflow net in a stepwise fashion by replacing nodes that represent high-level actions with workflow nets which describe these actions in more detail.

An advantage of such a hierarchical approach is that it produces workflow nets with an explicit and natural hierarchical structure, which considerably aids the understandability of the specification [7, 8]. It can for example be used in the design and analysis tool and allow the user to zoom in and out of specific parts of the net by either expanding or collapsing nodes according to the hierarchical structure. In addition the structure can often be matched with the organizational hierarchy of the organization that hosts the specified workflow, and therefore be linked with for example the levels of management.

The specific refinement approach that we will take in this paper works as follows. We always start with a simple type of net that we already know to have the desired soundness properties. Then we allow the substitution of a single node, either a place or a transition, with a workflow net that we also already know to possess the desired properties. We will show that for suitable soundness properties and specific types of substitutions it will hold that the soundness properties are preserved, i.e., the result of the substitution also has the soundness properties. This allows us to start from a small set of simple nets that are known to have the desired properties, and then from then generate a larger class of nets that also have these properties. This idea of net refinements is quite old, and the first papers were published in the early 90's, like [9]. Methods for stepwise refinements were studied in numerous papers, including [10], [11], [12] or [13]. An approach that we will in particular focus on is the one presented by van Hee et al. in [14] where two large classes of simple workflow nets, based on state machines and marked graphs, are identified which are readily observed to be sound, and then it is shown that when closed under substitution we get a larger class of workflow nets, called ST-nets, which contains also only sound workflow nets.

In another similar approach the substitution is used as follows. We always start with a net consisting of a single place, and allow only the substitution with a finite set of simple nets but which are not necessarily workflow nets themselves, such as for example by Wachtel et al. in [15] and van Hee et al. in [16]. Interestingly enough, this leads to a slightly different class being generated

than the one in [14], neither strictly larger nor smaller. It is the main goal of this paper to investigate the combination of these two approaches and see if it allows the generation of even larger classes of sound nets.

For the approach chosen in this paper we need a special notion of soundness. This is because, as was observed by van Hee et al. in [14], it is unfortunately in general not true that soundness as defined earlier is preserved by substitution, i.e., if we substitute a sound net in another sound net the result is not necessarily sound. This is related to the fact that although if we execute a sound workflow, starting with a single token, then we will end up with a single token in the output place and no other tokens anywhere, it could be that if we start the same workflow with 2 tokens, it does not necessarily mean that the final marking will have 2 tokens in the output place. It can therefore happen that substitution of such a workflow net will lead to an unsound net.

For this reason the notion of k -soundness was introduced by van Hee et al, where k is a parameter for which whenever we start with k tokens, the net will end without deadlock having exactly k tokens in the output place, while all other places will be unmarked. It was proven that k -soundness forms a strict hierarchy, which means that for every k there exist a workflow net which is k -sound and not $(k + 1)$ -sound. The notion of $*$ -soundness is reserved for nets, which are sound for every k . It is shown by van Hee et al. in [14], that this type of soundness is preserved by substitution for their kind of nets. In the same paper van Hee et al. define a large class of nets by starting from very simple classes that are syntactically easy to identify and can be straightforwardly shown to be $*$ -sound, and then generating more $*$ -sound nets by substitution.

Since in this paper we consider a more generalized notion of substitution that also allows substitution of nets with multiple input and output places and allows flow edges that arrive in input places and leave from output places, our approach requires a slightly generalized notion of soundness that we call *substitution soundness* and which is indeed preserved by the generalized type of substitution that we propose.

The structure of the paper is as follows. After introducing the notions of a Petri net, workflow net and soundness we propose a new classes of nets, called p-WF nets and t-WF nets. Such nets have the border nodes being places or transitions respectively. AND-OR nets being special classes of p-WF nets and t-WF nets are introduced in Section 3. We make some remarks on their properties and specify how the substitutions are performed. Next, we address the problem of soundness preservation during substitution in Section 4

and introduce the notion of *substitution-soundness* (sub-soundness for short). The main two theorems of this section state that soundness is preserved when a sub-sound t-WF net is substituted for a transition of a sub-sound p-WF net or t-WF net and when a sub-sound p-WF net is substituted for a place of a sub-sound p-WF net or t-WF net. In Section 5 we prove that the introduced AND-OR nets are sub-sound in general.

A preliminary version of this paper was presented in 2011 at the 11th International Conference on Application of Concurrency to System Design in Newcastle upon Tyne, United Kingdom, see [17]. Apart of providing a more elaborate discussion of the results, the main extensions in this paper include the complete versions of proofs and discussion on whether substitution soundness is the right notion of soundness, i.e., is the weakest condition necessary for constructing nets by refinement.

2. Basic Terminology

Let S be a set. A bag (multiset) m over S is a function $m : S \rightarrow \mathbb{N}$. We use $+$ and $-$ for the sum and the difference of two bags and $=, <, >, \leq, \geq$ for comparisons of bags, which are defined in a standard way. We overload the set notation, writing \emptyset for the empty bag and \in for the element inclusion. We list elements of bags between brackets, e.g. $m = [p^2, q]$ for a bag m with $m(p) = 2$, $m(q) = 1$, and $m(x) = 0$ for all $x \notin \{p, q\}$. The shorthand notation $k.m$ is used to denote the sum of k bags m . The size of a bag m over S is defined as $|m| = \sum_{s \in S} m(s)$.

Definition 1 (Petri net). A Petri net is a tuple $N = (P, T, F)$ with P a finite set of places, T a finite set of transitions such that $P \cap T = \emptyset$ and $F \subseteq (T \times F) \cup (F \times T)$ the set of flow edges.

A path of a net is a non-empty sequence (x_1, \dots, x_n) of nodes such that for all i such that $1 \leq i \leq n-1$ it holds that $(x_i, x_{i+1}) \in F$. Markings are states (configurations) of a net and the set of markings of $N = (P, T, F)$ is the set of all bags over P and denoted as \mathbf{M}_N . Given a transition $t \in T$, the preset $\bullet t$ and the postset $t \bullet$ of t are the sets $\{p \mid (p, t) \in F\}$ and $\{p \mid (t, p) \in F\}$, respectively. Analogously we write $\bullet p$, $p \bullet$ for pre- and postsets of places. To emphasize the fact that the preset/postset is considered within some net N , we write $\bullet_N t$, $a \bullet_N$. We overload this notation further allowing to apply preset and postset operations to a set B of places/transitions, which is defined as the union of pre-/postsets of elements of B . A transition $t \in T$ is said to be enabled in marking m iff $\bullet t \leq m$. For a net $N = (P, T, F)$ with markings m_1 and m_2 and a transition $t \in T$ we write

$m_1 \xrightarrow{t}_N m_2$ if t is enabled in m_1 and $m_2 = m_1 - \bullet t + t \bullet$. For a sequence of transitions $\sigma = \langle t_1, \dots, t_n \rangle$ we write $m_1 \xrightarrow{\sigma}_N m_n$ if $m_1 \xrightarrow{t_1}_N m_2 \xrightarrow{t_2}_N \dots \xrightarrow{t_n}_N m_n$, and we write $m_1 \xrightarrow{*}_N m_n$, if there exists such a sequence $\sigma \in T^*$. We will write $m_1 \xrightarrow{t} m_2$, $m_1 \xrightarrow{\sigma} m_n$ and $m_1 \xrightarrow{*} m_n$, if N is clear from the context.

We generalize the usual notion of workflow net as introduced by van der Aalst in [2] by allowing multiple input and output places, allowing transitions as input and output nodes and also allowing input nodes to have incoming edges and output nodes to have outgoing edges.

Definition 2 (Workflow net). A place Workflow net (pWF net) is a tuple $N = (P, T, F, I, O)$ where (P, T, F) is a Petri net with a non-empty set $I \subseteq P$ of input places and a non-empty set $O \subseteq P$ of output places such that (1) every node in $P \cup T$ is reachable by a path from at least one node in I and (2) from every node in $P \cup T$ we can reach at least one node in O . A transition Workflow net (tWF net) is similar to a place Workflow net except that I and O are non-empty subsets of T . A workflow net (WF net) is either a pWF net or tWF net.

A workflow net is called a *one-input* workflow net if I contains one element, and a *one-output* workflow net if O contains one element. In [2] workflow nets are restricted to one-input one-output place Workflow nets. We generalize this but define for all workflow nets the corresponding one-input one-output pWF net as follows. The *place-completion* of a tWF net $N = (P, T, F, I, O)$ is denoted as $\mathbf{pc}(N)$ and is a one-input one-output pWF net that is constructed from N by adding places p_i and p_o such that $p_i \bullet = I$ and $\bullet p_o = O$ and setting the input set and output set as $\{p_i\}$ and $\{p_o\}$ respectively. This is illustrated in Figure 1 (a). Note that we distinguish I nodes with half unconnected incoming arrows and O nodes with half unconnected outgoing arrow. The *transition-completion* of a pWF net $N = (P, T, F, I, O)$ is denoted as $\mathbf{tc}(N)$ and is a one-input one-output tWF net that is constructed from N by adding transitions t_i and t_o such that $t_i \bullet = I$ and $\bullet t_o = O$ and setting the input set and output set as $\{t_i\}$ and $\{t_o\}$, respectively. This is illustrated in Figure 1 (b).

We will focus in this paper on a particular kind of soundness, namely the soundness that guarantees the reachability of a proper final state. We generalize this for the case where there can be more than one input place and these contain one or more tokens in the initial marking. We also provide a generalization of soundness for tWF nets, which intuitively states that, if in total there are k firings of input transitions, then the computa-

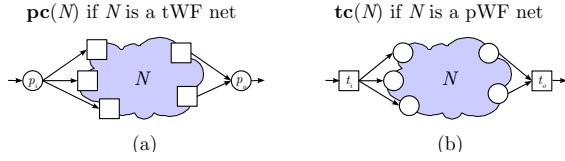


Figure 1: The place completion of a tWF net and a pWF net

tion will end in an empty marking after in total k firings of the output transitions.

Definition 3 (k and $*$ -soundness). A pWF net $N = (P, T, F, I, O)$ is said to be k -sound if for each marking m such that $k \cdot I \xrightarrow{*} m$ it holds that $m \xrightarrow{*} k \cdot O$. We call N $*$ -sound if it is k -sound for all $k \geq 1$. We say that these properties hold for tWF net N if they hold for $\mathbf{pc}(N)$.

It would be nice if transition-completion would not affect the $*$ -soundness of a net just like place-completion does (by definition). However this is only partially true as is shown in the following theorem.

Theorem 1. Every pWF net N is $*$ -sound if $\mathbf{tc}(N)$ is $*$ -sound but not vice versa.

PROOF. Let $N = (P, T, F, I, O)$ and $N' = \mathbf{pc}(\mathbf{tc}(N)) = (P', T', F', I', O')$ with t_i and t_o being the added input and output transitions of $\mathbf{tc}(N)$, respectively. Recall that by definition $\mathbf{tc}(N)$ is $*$ -sound iff $\mathbf{pc}(\mathbf{tc}(N))$ is $*$ -sound. We assume that $\mathbf{tc}(N)$ is $*$ -sound, that is N' is $*$ -sound. Observe that $k \cdot I' \xrightarrow{*}_{N'} k \cdot I$ by letting input transitions t_i of $\mathbf{tc}(N)$ fire k times. Assume that $k \cdot I \xrightarrow{*}_N m$. Since N is embedded in N' it then follows that $k \cdot I' \xrightarrow{*}_{N'} m$. From the $*$ -soundness of N' it follows that $m \xrightarrow{\sigma'}_{N'} k \cdot O'$ for some $\sigma' \in (T')^*$. However, we can omit the firings of t_o from σ' and obtain σ such that $m \xrightarrow{\sigma}_{N'} k \cdot O$. Since σ cannot contain t_i it follows that $m \xrightarrow{\sigma}_N k \cdot O$ and therefore $m \xrightarrow{*}_N k \cdot O$.

The counterexample in Figure 2 shows that not for every $*$ -sound pWF net N it holds that $\mathbf{tc}(N)$ is $*$ -sound. Observe that N is $*$ -sound. However the shown $\mathbf{pc}(\mathbf{tc}(N))$ is not since from the marking $[p_i]$ it can reach $[b, c]$ and therefore $[b, p_o]$ after which no transition is enabled. Since $\mathbf{pc}(\mathbf{tc}(N))$ is not 1-sound, then by definition $\mathbf{tc}(N)$ is also not 1-sound and thus not $*$ -sound.

3. AND-OR nets

To generate a large class of nets we will consider general substitutions where places and transitions are replaced with pWF nets and tWF nets, respectively. We

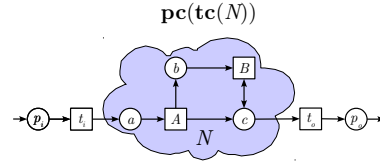


Figure 2: A counterexample showing that $*$ -soundness is not preserved by transition completion and also not under substitution

introduce for this purpose a notion of substitution that is based on the one introduced by van Hee et al. in [14] but generalized so it can substitute nets with multiple input nodes and multiple output nodes.

Definition 4 (Place substitution, Transition substitution).

Consider two disjoint WF nets N and M , i.e., if $N = (P, T, F, I, O)$ and $M = (P', T', F', I', O')$ then $(P \cup T) \cap (P' \cup T') = \emptyset$.

Place substitution: If p is a place in N and M is a pWF net, then we define the result of substituting p in N with M , denoted as $N \otimes_p M$, as the net that is obtained if in N we remove p and the edges in which it participates and replace it with the net M and edges such that $\bullet p' = \bullet p$ for each input place $p' \in I'$ of M and $p' \bullet = p \bullet$ for each output place $p' \in O'$ of M . If $p \in I$ then p is replaced in the set of input nodes of the resulting net with I' , i.e., the input set of $N \otimes_p M$ is $(I \setminus \{p\}) \cup I'$, and if $p \in O$ then p is replaced in the set of output nodes of the resulting net with O' , i.e., the output set of $N \otimes_p M$ is $(O \setminus \{p\}) \cup O'$. Otherwise, the input and output sets of $N \otimes_p M$ are the same as the respective sets for N .

Transition substitution: Likewise, if t is a transition in N and M is a tWF net, then we define the result of substituting t in N with M , denoted as $N \otimes_t M$, as the net that is obtained if in N we remove t and the edges in which it participates and replace it with the net M and edges such that $\bullet t' = \bullet t$ for each input transition $t' \in I'$ of M and $t' \bullet = t \bullet$ for each output transition $t' \in O'$ of M . If $t \in I$ then t is replaced in the set of input nodes of the resulting net with I' , i.e., the input set of $N \otimes_t M$ is $(I \setminus \{t\}) \cup I'$, and if $t \in O$ then t is replaced in the set of output nodes of the resulting net with O' , i.e., the output set of $N \otimes_t M$ is $(O \setminus \{t\}) \cup O'$. Otherwise, the input and output sets of $N \otimes_t M$ are the same as the respective sets for N .

The results of a place substitution and transition substitution are illustrated in Figure 3 (a) and (b), respectively. It is not hard to see that if N and M are WF nets and n a node in N then $N \otimes_n M$ is again a WF net. It also holds for all WF nets A , B and C that $(A \otimes_a B) \otimes_b C = A \otimes_a (B \otimes_b C)$ if b is a node in B ,

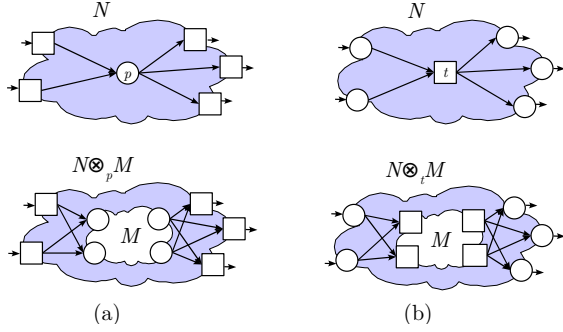


Figure 3: Illustration of place substitution and transition substitution

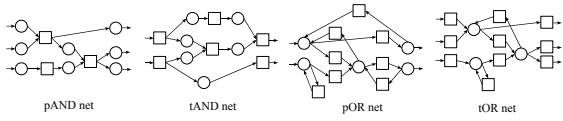


Figure 4: Examples of a pAND, tAND, pOR and tOR nets

and $(A \otimes_a B) \otimes_b C = (A \otimes_a C) \otimes_b B$ if a and b are nodes in A .

We will generate nets by starting from some basic classes of nets and allowing substitutions of places with pWF nets and transitions with tWF nets.

Definition 5 (Substitution closure). Given a class C of nets we defined the substitution closure of C , denoted as $\mathbf{S}(C)$, as the smallest superclass of C that is closed under transition substitution and place substitution, i.e., the following two rules hold: if N and M are disjoint nets in $\mathbf{S}(C)$ then (1) if M is a pWF net and p a place in N then $N \otimes_p M$ is a net in $\mathbf{S}(C)$ and (2) if M is a tWF net and t a transition in N then $N \otimes_t M$ is a net in $\mathbf{S}(C)$.

As the basic nets with which we will start the generation process we will consider the nets that we call pAND nets, tAND nets, pOR nets and tOR nets, which are all illustrated in Figure 4 with input and output nodes on the left-hand side and right-hand side, respectively. Informally we can describe AND nets as acyclic nets that consist only of AND splits and AND joins, and OR nets can be described as possibly cyclic nets consisting of only OR splits and OR joins. AND and OR nets are generalizations of marked graph/T-nets and state machines/S-nets [6], respectively, which both are restricted to exactly one input and output node. More formally, the AND and OR nets are defined as follows.

Definition 6 (AND net). An AND net is an acyclic WF net (P, T, F, I, O) such that for every place $p \in P$ it holds

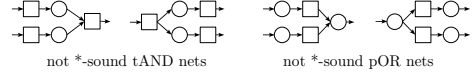


Figure 5: Examples of tAND and pOR nets that are not *-sound

that (1) $p \in I \wedge |\bullet p| = 0$ or $p \notin I \wedge |\bullet p| = 1$ and (2) $p \in O \wedge |p \bullet| = 0$ or $p \notin O \wedge |p \bullet| = 1$. An AND net that is a pWF net is called a pAND net, and if it is a tWF net it is called a tAND net.

OR nets are the counterpart of AND nets and are defined as follows.

Definition 7 (OR net). An OR net is a WF net (P, T, F, I, O) such that for every transition $t \in T$ it holds that (1) $t \in I \wedge |t \bullet| = 0$ or $t \notin I \wedge |t \bullet| = 1$ and (2) $t \in O \wedge |t \bullet| = 0$ or $t \notin O \wedge |t \bullet| = 1$. An OR net that is a pWF net is called a pOR net, and if it is a tWF net it is called a tOR net.

Note that OR nets can contain cycles where AND nets by definition cannot, but otherwise they are each others dual. Also note that for the requirements over the edges, being an input node counts as having an input edge, and being an output node counts as having an output edge. To illustrate why this is so consider the nets (a) and (b) in Figure 6. In (a) we see a pWF net that would be a pAND net if we ignored the requirement for input and output places. However, it will also be clear that this is not a 1-sound net since the token in the upper-right output place might be transferred to the lower-left input place, after which we cannot reach the final state. In (b) we see a tWF net that would be tOR net if we ignored the requirements for input and output places. Also here it is easy to see by looking at its place completion that this is not a 1-sound net. For tAND and pOR there are no such restrictions on the input and output nodes, since in AND nets the places are restricted and in OR nets the transitions are restricted. The requirement for acyclicity for AND nets is illustrated by the tWF net (c) in Figure 6. Clearly this net is not 1-sound since a run in which the transition fires requires an initial token in the place. However, its dual where the place is a transition and vice versa, is indeed 1-sound, which explains the asymmetry between AND and OR nets.

For the AND and OR nets as defined here there are some straightforward soundness results in that all pAND and tOR nets are *-sound, and that for tAND and pOR nets this is the case if they are one-input one-output nets. The *-soundness of tOR nets follows from

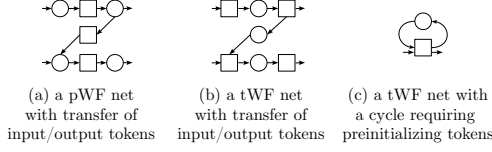


Figure 6: Unsound nets forbidden by the definition of AND and OR nets

the $*$ -soundness of ST-nets of van Hee et al. given by Theorem 17 in [14] and the definition of $*$ -soundness for tWF nets by place completion. The $*$ -soundness of pAND nets follows from Theorem 1 and the fact that for every pAND net N , its transition completion $\mathbf{tc}(N)$ is $*$ -sound because $\mathbf{pc}(\mathbf{tc}(N))$ is also an ST-net. Note that for this reasoning it is crucial that the place completion of a tOR net results in a net that is still an OR net, and that transition completion of a pAND net results in a net that is still an AND net. This is the case since in tOR nets and in pAND nets input nodes cannot have incoming edges and output nodes cannot have outgoing edges.

Observe that even without disallowing incoming edges for input nodes and outgoing edges for output nodes, a place completion of any tAND net results in an AND net and a transition completion of any pOR net results in OR net, i.e., results in nets that do not have AND splits/joins and OR splits/joins intermixed in a problematic way. Note also that for multi-input multi-output pOR nets an unsound transfer would be possible similarly as for pAND nets, but we limit the number of input/output places anyway. Finally, even though for tAND nets we do not limit the number of these edges in the definition, it follows from its acyclicity and existence of only one input and one output transition.

To understand the restriction to one-input one-output nets consider the examples of tAND and pOR nets in Figure 5 which are all nets with either multiple input nodes or multiple output nodes and which are all not $*$ -sound. For the presented tAND net examples applying the place completion, which is required by the definition of soundness, would result in a net with AND splits/joins and OR splits/joins mixed in a wrong way. For the presented pOR net examples the problem originates from the nature of allowed OR splits/joins and the possibility of unequal numbers of input and output places. This is why, while generating nets with place and transition substitution, we limit ourselves to the following classes of nets: the class of pAND nets represented by **pAND**, the class of one-input one-output tAND nets represented by **11tAND**, the class of one-input one-output pOR nets represented by **11pOR**, and

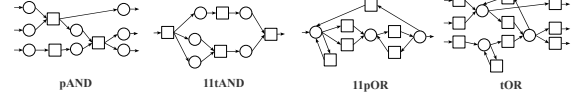


Figure 7: Example nets from classes **pAND**, **11tAND**, **11pOR** and **tOR**

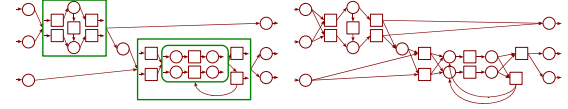


Figure 8: An example of the generation of an AND-OR net

the class of tOR nets represented by **tOR** (see Figure 7 for examples). For one-input one-output tAND nets the $*$ -soundness follows immediately from the $*$ -soundness of pAND nets because performing place completion of one-input one-output tWF nets does not create OR splits nor OR joins. For one-input one-output pOR nets we cannot refer to $*$ -soundness of ST-nets, because they cannot have incoming edges for input places and outgoing edges for output places. Yet, by its construction the number of tokens in the net has to be constant and by reachability of input and output nodes in the definition of workflow net all tokens can be forced to reach the output place. Section 5 provides formal proofs of stronger sub-soundness properties for all the basic classes discussed here.

Definition 8 (AND-OR net). The class $\mathbf{S}(\mathbf{pAND} \cup \mathbf{11tAND} \cup \mathbf{11pOR} \cup \mathbf{tOR})$ we call the class of AND-OR nets.

An example of the generation of an AND-OR net is shown in Figure 8, with on the left-hand side the hierarchical decomposition and on the right-hand side the resulting net.

It can be shown that the one-input one-output tAND nets are not needed, i.e., we can remove them from the initial class without changing the set of nets that can be generated.

Theorem 2. *The tAND nets are redundant for generating AND-OR nets, i.e., $\mathbf{S}(\mathbf{pAND} \cup \mathbf{11tAND} \cup \mathbf{11pOR} \cup \mathbf{tOR}) = \mathbf{S}(\mathbf{pAND} \cup \mathbf{11pOR} \cup \mathbf{tOR})$.*

PROOF. Recall that tAND nets do not contain cycles. Also note that if we take a one-input one-output tAND

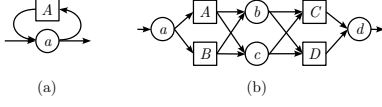


Figure 9: Examples showing the expressive power of certain classes

net with input transition t_i and output transition t_o and we remove the begin and end transition, then we are left with a pAND net with $I = t_i \bullet$ and $O = \bullet t_o$. So every one-input one-output tAND net can be generated by starting with an tOR net consisting of a transition followed by a place which is again followed by a transition, and then substituting the previously mentioned pAND net for the place in the middle.

However, the one-input one-output pOR nets are not redundant, because a cycle containing the input and output nodes cannot be obtained in any other way.

Theorem 3. *The pOR nets are not redundant for generating all AND-OR nets, i.e., $S(\text{pAND} \cup \mathbf{1tAND} \cup \mathbf{1pOR} \cup \mathbf{tOR}) \supsetneq S(\text{pAND} \cup \mathbf{1tAND} \cup \mathbf{tOR})$.*

PROOF. See the counterexample in Figure 9 (a). This one-input one-output pOR net cannot be generated by using pAND, one-input one-output tAND and tOR nets.

Of course pAND nets and tOR nets are not redundant either, since they allow for multiple input and output nodes.

The AND-OR nets are very similar to the ST nets defined in [14] by van Hee et al. In fact, the class of ST nets is the strict subclass of $S(\mathbf{1tAND} \cup \mathbf{1pOR})$ that disallows incoming edges for input nodes and outgoing edges for output nodes. It is clear that the class $S(\mathbf{1tAND} \cup \mathbf{1pOR})$ is a proper subclass of the AND-OR nets since it only contains one-input one-output WF nets. However there are in addition also one-input one-output AND-OR nets that are not in $S(\mathbf{1tAND} \cup \mathbf{1pOR})$ as is shown by the following theorem.

Theorem 4. *The class $S(\mathbf{1tAND} \cup \mathbf{1pOR})$ does not contain all one-input one-output AND-OR nets.*

PROOF. The counterexample is given in Figure 9 (b). To show that it is an AND-OR net we consider its generation in reverse. The transitions A and B form an tOR net and can be contracted into a single transition. The same for the transitions C and D. The places b and c form a pAND net and can be contracted into a single

place. The result will be a linear net that is in fact both a pAND net and a one-input one-output pOR net. To see that the example net is not in $S(\mathbf{1tAND} \cup \mathbf{1pOR})$ it can be verified that there is no proper subnet that is either in $\mathbf{1tAND}$ or $\mathbf{1pOR}$ and can be contracted into a single transition or place, respectively.

4. Substitution soundness

Recall that the purpose of this paper is to investigate the possibility to generate a large class of *-sound WF nets by using substitution. It is unfortunately not true that *-soundness is preserved by substitutions as defined in this paper. This is because of the possible outgoing edges of the output nodes. A counterexample is shown in Figure 2 where the presented pWF net can be thought of as being constructed by substituting a *-sound net N , with input place a and output place c , into an also *-sound sequential pWF net. As was discussed in the proof of Theorem 1 the resulting net is not 1-sound so also not *-sound. Therefore, we introduce a new notion of soundness called *substitution soundness* and study its properties. As we will show in Section 5 that all the basic classes of nets from the definition of AND-OR nets are substitution sound.

The intuition underlying substitution soundness is that it should not matter that during a run of a workflow net we remove seemingly ready tokens from output places. In other words, it should hold that if the net starts with k tokens in the input places, reaches a marking with at least $k' \leq k$ tokens in each output place, and we remove these k' tokens from each output place, then the net can still finish with $k - k'$ tokens in each output place.

Definition 9 (Substitution soundness). Let

$N = (P, T, F, I, O)$ be a pWF net. We say that N is *substitution-sound* (or simply *sub-sound*) if for all $k \geq k' \geq 0$ and every marking m' it holds that if $k \cdot I \xrightarrow{*} (m' + k' \cdot O)$ then $m' \xrightarrow{*} (k - k') \cdot O$. We generalize this property to tWF nets and say that a tWF net N is sub-sound if $\text{pc}(N)$ is sub-sound.

We claim this is in some sense a necessary condition to construct 1-sound nets by substitution of nodes in 1-sound nets. In particular it can be shown that there is no weaker condition that is preserved by substitution and implies 1-soundness.

Theorem 5. *There is no property of pWF nets that (1) is strictly weaker than substitution soundness, i.e., it is implied by substitution soundness but not vice versa, (2)*

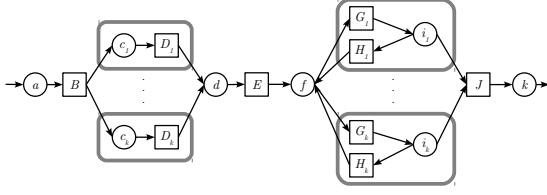


Figure 10: Illustration of necessity of substitution soundness

implies 1-soundness and (3) is preserved by substitution.

PROOF. Consider the class of pWF nets illustrated in Figure 10 where a pWF net is defined for each value of k , which we will call M_k . It can be easily observed that these nets are 1-sound, and in fact are substitution sound. Now consider a pWF net N that is not substitution sound such that if we let it start with k tokens in the input places and during its run remove $k' < k$ tokens from the output places then it cannot reach the final marking. If we substitute N in M_k for place d , i.e., we consider $M_k \otimes_d N$, then we obtain a net that is not 1-sound. To see this consider the following. We can let M_k start with one token in a and run until there are k tokens in the input places of N . Then we can run N until there are k' tokens in its output places. These tokens can then be removed by firing k times E . Since after this N cannot reach a final state with $k - k'$ in its output places, it follows that the net $M_k \otimes_d N$ cannot reach its final state.

The theorem now follows from the previous by the following reduction ad absurdum. Assume some property that satisfies (1), (2) and (3) at the same time. Observe that M_k will satisfy this property since this property is weaker than substitution soundness. Also observe that there has to be pWF net N that satisfies this property but is not substitution sound. By (3) it then follows that $M_k \otimes_d N$ also has the property and therefore by (2) that it is 1-sound. This, however, contradicts what we observed previously, namely that the result is not 1-sound.

Observe that the previous theorem does not establish that substitution soundness is necessary in the sense that every property that satisfies the three conditions is stronger than substitution soundness. This is however the case if we add the requirement that that the property must hold for the nets in Figure 10, i.e., these nets should be in the class of nets that we intend to generate by substitution, which seems a reasonable requirement.

Now we prove that sub-soundness is sufficient for constructing *-sound nets by substitution. First, note that the case where $k' = 0$ describes *-soundness and so sub-soundness implies *-soundness. Furthermore, on many classes of nets the two notions of soundness coincide, as is shown by the following two lemmas.

Lemma 6. *For every pWF net N such that all output places have no outgoing edges it holds that N is *-sound iff N is sub-sound.*

PROOF. As already argued it holds that sub-soundness implies *-soundness, so the converse remains to be shown. Let $N = (P, T, F, I, O)$. Assume that $k.I \xrightarrow{*} (m + k'.O)$ for some k' such that $k \geq k' \geq 0$. By *-soundness it holds for some σ that $(m + k'.O) \xrightarrow{\sigma} k.O$. However, since the places in O have no outgoing edges none of the transitions in σ consumes any of their tokens and so $m \xrightarrow{\sigma} (k - k').O$.

Note that the restriction mentioned in Lemma 6 is included in the classical definition of WF net by van der Aalst [2]. However, with this restriction we would not be able to generate all AND-OR nets, not even all those that satisfy this restriction. In particular we would not be able to do arbitrary loop additions. As an example consider Figure 9 (b) where we would not be able to add a loop to place b . Note that a similar restriction is not necessary for tWF nets because for them the soundness properties are defined by place completion. Recall also that for tOR nets the output transitions cannot have outgoing edges by definition and for one-input one-output tAND nets this follows from the facts that AND nets are acyclic and that in a tWF nets it is possible to reach one of the output transitions from every place and transition.

Lemma 7. *For every tWF net N it holds that N is *-sound iff N is sub-sound.*

PROOF. As already argued, it is enough to show that *-soundness implies sub-soundness. A tWF net N is by definition sub-sound iff $\mathbf{pc}(N)$ is sub-sound. Since in $\mathbf{pc}(N)$ the output place has no outgoing edges it follows from Lemma 6 that $\mathbf{pc}(N)$ is sub-sound iff it is *-sound. Finally, by definition it holds that $\mathbf{pc}(N)$ is *-sound iff N is *-sound.

We now proceed with showing that sub-soundness is sufficient for constructing *-sound nets by substitution. In Theorems 8 and 10 we show that sub-soundness is preserved while substituting places in pWF nets and tWF nets respectively, while in Theorems 14 and 15 we show that is also preserved while substituting transitions in pWF nets and tWF nets respectively.

Theorem 8. If a pWF net $N = (P_N, T_N, F_N, I_N, O_N)$ and a disjoint pWF net $M = (P_M, T_M, F_M, I_M, O_M)$ are sub-sound, then for any $p \in P_N$ the net $N \otimes_p M$ is also sub-sound.

PROOF. Let $N_{NM} = N \otimes_p M = (P_{NM}, T_{NM}, F_{NM}, I_{NM}, O_{NM})$. We define $\mathbf{S}(M, k)$ as the set of markings m_M of M that represent the fact that there are still k “threads” active in M after possibly having started with more threads but some of them ended by the removal of tokens from O' , i.e., for some $k' \geq k$ it holds that $k'.I' \xrightarrow{*}_M m_M + (k' - k).O_M$. We define a simulation relation $\sim \subseteq \mathbf{M}_N \times \mathbf{M}_{NM}$ such that $m_N \sim m_{NM}$ represents the fact that m_N is the same as m_{NM} except that all (say k) tokens are removed from p and replaced by some marking from $\mathbf{S}(M, k)$, i.e., $m_{NM} = m_N - [p^k] + m_M^k$ for some $m_M^k \in \mathbf{S}(M, k)$ with $k = m_N(p)$.

We first discuss the idea of the proof and then follow with the laborious details.

It can be shown that \sim indeed defines a kind of bisimilarity, i.e., (see Figure 11) it holds that (1*) if $m_N \xrightarrow{\sigma}_N m'_N$ and $m_N \sim m_{NM}$, then there is a marking $m'_{NM} \in \mathbf{M}_{NM}$ such that $m_{NM} \xrightarrow{*}_{NM} m'_{NM}$ and $m'_N \sim m'_{NM}$ and (2*) if $m_{NM} \xrightarrow{\sigma}_{NM} m'_{NM}$ and $m_N \sim m_{NM}$, then there is a marking $m'_N \in \mathbf{M}_N$ such that $m_N \xrightarrow{*}_N m'_N$ and $m'_N \sim m'_{NM}$. This can be shown with induction on the length of σ where for each transition t in σ we distinguish for (1*) the cases where $p \in \bullet_N t$ or not and $p \in t \bullet_N$ or not, and for (2*) we distinguish the cases where t is a transition in N or M .

We then can show the sub-soundness of $N \otimes_p M$ using (1*) and (2*). The idea of this part is as follows. Assume that $k.I_{NM} \xrightarrow{*}_{NM} (m_{NM} + k'.O_{NM})$ with $k \geq k' \geq 0$. By (2*) and the fact that $k.I_N \sim k.I_{NM}$ it then follows that $k.I_N \xrightarrow{*}_N m_N$ such that $m_N \sim (m_{NM} + k'.O_{NM})$. We can show that we can assume that $m_N = m'_N + k'.O_N$ with m'_N a marking of N . By the sub-soundness of N it holds that $m'_N \xrightarrow{*}_N (k - k').O_N$. We can show that $m'_N \sim m_{NM}$ and so from (1*) it then follows that $m_{NM} \xrightarrow{*}_{NM} m'_{NM}$ such that $(k - k').O_N \sim m'_{NM}$. Although similar to $(k - k').O_N$, the m'_{NM} does not have to be the final marking $(k - k').O_{NM}$, yet by using sub-soundness of M it can be shown that $m'_{NM} \xrightarrow{*}_{NM} (k - k').O_{NM}$.

We proceed with the proof of (1*) and (2*). We first show that (0) if $m_N \sim m_{NM}$, $m_N \xrightarrow{t}_N m'_N$ and $m_{NM} \xrightarrow{t}_{NM} m'_{NM}$, then $m'_N \sim m'_{NM}$. We then use (0) to show (1) if $m_N \sim m_{NM}$ and $m_N \xrightarrow{t}_N m'_N$, then there is a marking m'_{NM} such that $m_{NM} \xrightarrow{*}_{NM} m'_{NM}$ and $m_{NM} \sim$

m'_{NM} , and (2) if $m_N \sim m_{NM}$ and $m_{NM} \xrightarrow{t}_{NM} m'_{NM}$, then there is a marking m'_N such that $m_N \xrightarrow{*}_N m'_N$ and $m'_N \sim m'_{NM}$. Then with induction we generalize (1) and (2) to (1*) and (2*), respectively.

We show (0) as follows. Assume that $m_N \xrightarrow{t}_N m'_N$ and $m_{NM} \xrightarrow{t}_{NM} m'_{NM}$. We also assume $m_N \sim m_{NM}$, which by definition gives $m_{NM} = m_N - [p^k] + m_M^k$ for some $m_M^k \in \mathbf{S}(M, k)$ with $k = m_N(p)$. After firing t in m_{NM} we get $m'_{NM} = (m_N - [p^k] + m_M^k - \bullet_{NM} t + t \bullet_{NM})$. We consider the four cases for whether $p \in \bullet_N t$ or not, and $p \in t \bullet_N$ or not:

(i) Assume $p \notin \bullet_N t$ and $p \notin t \bullet_N$. In that case $\bullet_{NM} t = \bullet_N t$ and $t \bullet_{NM} = t \bullet_N$ and therefore $m'_{NM} = (m_N - \bullet_N t + t \bullet_N - [p^k] + m_M^k)$ and since $m_N \xrightarrow{t}_N m'_N$ it follows that $m'_{NM} = (m'_N - [p^k] + m_M^k)$. Now, it remains to be shown that $m'_N(p) = k$ which follows from $m_N(p) = k$ and $p \notin \bullet_N t$ and $p \notin t \bullet_N$. This concludes that $m'_N \sim m'_{NM}$.

(ii) Assume $p \in \bullet_N t$ and $p \notin t \bullet_N$. In that case $t \bullet_{NM} = t \bullet_N$ and from the construction of the substitution it follows that $\bullet_{NM} t = \bullet_N t - [p] + O_M$ and therefore $m'_{NM} = (m_N - \bullet_N t + t \bullet_N - [p^{(k-1)}] + m_M^k - O_M)$ and since $m_N \xrightarrow{t}_N m'_N$ it follows that $m'_{NM} = (m'_N - [p^{(k-1)}] + m_M^k - O_M)$. Then, it holds that (a) $m'_N(p) = k - 1$ because $m_N(p) = k$ and $p \in \bullet_N t$ and $p \notin t \bullet_N$, and (b) $m_M^k - O_M \in \mathbf{S}(M, k - 1)$ since $m_M^k \in \mathbf{S}(M, k)$. Observe that $m_M^k - O_M$ is a valid state, i.e., there is a non-negative number of tokens in each place, because we assumed t is enabled in m_{NM} as well $p \in \bullet_N t$ and the m_M^k component of m_{NM} covers tokens in places from P_M . From (a) and (b) it follows that $m'_N \sim m'_{NM}$.

(iii) Assume $p \notin \bullet_N t$ and $p \in t \bullet_N$. In that case $\bullet_{NM} t = \bullet_N t$ and from the construction of the substitution $t \bullet_{NM} = t \bullet_N - [p] + I_M$ and therefore $m'_{NM} = (m_N - \bullet_N t + t \bullet_N - [p^{(k+1)}] + m_M^k + I_M)$ and since $m_N \xrightarrow{t}_N m'_N$ it follows that $m'_{NM} = (m'_N - [p^{(k+1)}] + m_M^k + I_M)$. Then, it holds that (a) $m'_N(p) = k + 1$ because $m_N(p) = k$ and $p \notin \bullet_N t$ and $p \in t \bullet_N$, and (b) $m_M^k + I_M \in \mathbf{S}(M, k + 1)$ since $m_M^k \in \mathbf{S}(M, k)$. From (a) and (b) it follows that $m'_N \sim m'_{NM}$.

(iv) Assume $p \in \bullet_N t$ and $p \in t \bullet_N$. In that case $\bullet_{NM} t = \bullet_N t - [p] + O_M$ and $t \bullet_{NM} = t \bullet_N - [p] + I_M$ and therefore $m'_{NM} = (m_N - \bullet_N t + t \bullet_N - [p^k] + m_M^k - O_M + I_M)$ and since $m_N \xrightarrow{t}_N m'_N$ it follows that $m'_{NM} = (m'_N - [p^k] + m_M^k - O_M + I_M)$. Then, it holds that (a) $m'_N(p) = k$ because $m_N(p) = k$ and $p \in \bullet_N t$ and $p \in t \bullet_N$, and (b) $m_M^k - O_M + I_M \in \mathbf{S}(M, k)$ since $m_M^k \in \mathbf{S}(M, k)$. Observe that $m_M^k - O_M + I_M$ is a valid state for the same reasons as in (ii). From (a) and (b) it follows that $m'_N \sim m'_{NM}$.

We have now covered all possible cases and in each of them concluded that $m'_N \sim m'_{NM}$ which finishes the

$$\begin{array}{ccc}
\begin{array}{c} m_N \xrightarrow{\sigma} m'_N \\ \sim \\ m_{NM} \end{array} & \xRightarrow{\exists m'_{NM}} & \begin{array}{c} m_N \xrightarrow{\sigma} m'_N \\ \sim \\ m_{NM} \xrightarrow{*} m'_{NM} \end{array} \\
(1^*) & &
\end{array}
\qquad
\begin{array}{ccc}
\begin{array}{c} m_N \\ \sim \\ m_{NM} \xrightarrow{\sigma} m'_{NM} \end{array} & \xRightarrow{\exists m'_N} & \begin{array}{c} m_N \xrightarrow{*} m'_N \\ \sim \\ m_{NM} \xrightarrow{\sigma} m'_{NM} \end{array} \\
(2^*) & &
\end{array}$$

Figure 11: \sim indeed defines a kind of bisimilarity

proof of (0).

We show (1) as follows. Assume that $m_N \xrightarrow{t} m'_N$, which by definition gives $\bullet_N t \leq m_N$. We also assume $m_N \sim m_{NM}$, which by definition gives $m_{NM} = m_N - [p^k] + m_M^k$ for some $m_M^k \in \mathbf{S}(M, k)$ with $k = m_N(p)$. Since $m_M^k \in \mathbf{S}(M, k)$ and M is sub-sound, it holds that $m_M^k \xrightarrow{*} k.O_M$, and since M is embedded in NM , it follows that $m_{NM} \xrightarrow{*}_{NM} m_N - [p^k] + k.O_M$. Note also that $m_{NM} \sim m_N - [p^k] + k.O_M$, because by definition $k.O_M \in \mathbf{S}(M, k)$. Now we observe that since t is a transition in N , if t is enabled in m_N for N , by construction of NM it is also enabled in $m_N - [p^k] + k.O_M$ for NM regardless of $p \in \bullet_N t$. It follows that there is a marking m'_{NM} such that $m_N - [p^k] + k.O_M \xrightarrow{t}_{NM} m'_{NM}$ and thus $m_{NM} \xrightarrow{*}_{NM} m'_{NM}$. By (0) it follows that $m'_N \sim m'_{NM}$, which concludes the proof of (1).

We show (2) as follows. Assume that $m_{NM} \xrightarrow{t}_{NM} m'_{NM}$. We also assume $m_N \sim m_{NM}$, which by definition gives $m_{NM} = m_N - [p^k] + m_M^k$ for some $m_M^k \in \mathbf{S}(M, k)$ with $k = m_N(p)$. We consider the two possible cases: t is a transition in N , and t is a transition in M .

(i) Assume that t is a transition in N . Since t was enabled in m_{NM} for NM , i.e., $\bullet_{NM} t \leq m_{NM}$, it will also be enabled in m_N for N , i.e., $\bullet_N t \leq m_N$. This can be shown as follows. Suppose $p \notin \bullet_N t$, then $\bullet_{NM} t = \bullet_N t$. Since m_M^k contains only places in M it follows from $\bullet_{NM} t \leq m_{NM} = m_N - [p^k] + m_M^k$ that $\bullet_N t = \bullet_{NM} t \leq m_N$. Suppose on the other hand that $p \in \bullet_N t$, then $\bullet_{NM} t = \bullet_N t - [p] + O_M$ and we get $\bullet_N t - [p] + O_M \leq m_N - [p^k] + m_M^k$. Both sides of this inequality can be limited to N by omitting components not from P_N , giving $\bullet_N t - [p] \leq m_N - [p^k]$. Since in this case $k \geq 1$ we get $\bullet_N t \leq m_N$. Now, since t is enabled in m_N for N there will be a marking m'_N such that $m_N \xrightarrow{t} m'_N$ and it follows by (0) that $m'_N \sim m'_{NM}$.

(ii) Assume that t is a transition in M . In this case the marking we are looking for is m_N itself. Since $\bullet_{NM} t$ are all places in M , it follows that t is enabled in m_M^k for M . So there is m'_M such that $m_M^k \xrightarrow{t}_M m'_M$ and because $t \bullet_{NM}$ are also all places in M , we have $m'_M = m_M^k - \bullet_{Mt} + t \bullet_M$. Now from our assumptions it follows that $m'_{NM} = m_{NM} - \bullet_{Mt} + t \bullet_M = m_N - [p^k] + m_M^k -$

$\bullet_{Mt} + t \bullet_M = m_N - [p^k] + m'_M$. Since $m_M^k \in \mathbf{S}(M, k)$ and $m_M^k \xrightarrow{t}_M m'_M$, then it also holds that $m'_M \in \mathbf{S}(M, k)$. From the assumption that $k = m_N(p)$, it follows that $m_N \sim m'_{NM}$, and obviously it also holds that $m_N \xrightarrow{*}_N m'_N$.

Since in both possible cases it follows that there is a marking m'_N such that $m_N \xrightarrow{*}_N m'_N$ and $m'_N \sim m'_{NM}$, we can conclude that this always follows, which concludes the proof of (2).

We can generalize (1) and (2) by using induction on the length of σ as follows: (1*) if $m_N \xrightarrow{\sigma}_N m'_N$ and $m_N \sim m_{NM}$ then there is a marking m'_{NM} such that $m_{NM} \xrightarrow{*}_{NM} m'_{NM}$ and $m'_N \sim m'_{NM}$ and (2*) if $m_{NM} \xrightarrow{\sigma}_{NM} m'_{NM}$ and $m_N \sim m_{NM}$, then there is a marking m'_N such that $m_N \xrightarrow{*}_N m'_N$ and $m'_N \sim m'_{NM}$.

We proceed with the proof of the final part, i.e., prove the sub-soundness of $N \otimes_p M$ using (1*) and (2*). The full structure of the reasoning is presented in Figure 12. Assume that $k.I_{NM} \xrightarrow{*}_{NM} (m_{NM} + k'.O_{NM})$ with $k \geq k' \geq 0$. Since $I_{NM} = I_N$ if $p \notin I_N$ and $I_{NM} = I_N - [p] + I_M$ if $p \in I_N$, it holds that $k.I_N \sim k.I_{NM}$. By (2*) it then follows that $k.I_N \xrightarrow{*}_N m_N$ such that $m_N \sim (m_{NM} + k'.O_{NM})$.

We now construct $m'_N = m_N - k'.O_N$ and show that $m'_N \sim m_{NM}$ regardless of $p \notin O_N$ or $p \in O_N$. We start with showing the fact that m'_N is a valid state, i.e., m_N includes the tokens we are subtracting from it. Since $m_N \sim (m_{NM} + k'.O_{NM})$ for $k'' = m_N(p)$ there is $m_M^{k''} \in \mathbf{S}(M, k'')$ such that $m_{NM} + k'.O_{NM} = m_N - [p^{k''}] + m_M^{k''}$. This gives $m_N = m_{NM} + k'.O_{NM} + [p^{k''}] - m_M^{k''}$. Let us consider two cases. For $p \notin O_N$, in which case $O_{NM} = O_N$, this gives $m_N = m_{NM} + k'.O_N + [p^{k''}] - m_M^{k''}$. It remains to observe that subtracting the $m_M^{k''}$ component does not remove any tokens from O_N because from disjointness of N and M we have $O_N \cap P_M = \emptyset$. For $p \in O_N$, in which case $O_{NM} = O_N - [p] + O_M$, we get $m_N = m_{NM} + k'.O_N + [p^{k''}] - [p^k] + k'.O_M - m_M^{k''}$. Both sides of the equality have to include the same number of tokens in p . Since m_{NM} marks only places from $P_{NM} = (P_N \setminus \{p\}) \cup P_M$ and $k'.O_M - m_M^{k''}$ only places from P_M (and $p \notin P_M$), all the tokens in p are given by $k'.O_N + [p^{k''}] - [p^k]$. It remains to show that $k'' \geq k'$.

$$\begin{array}{c}
\begin{array}{ccc}
k.I_{NM} \xrightarrow{*} m_{NM} + k'.O_{NM} & \xrightarrow{(2*)} & k.I_{NM} \xrightarrow{*} m_{NM} + k'.O_{NM} \\
\sim & & \sim \\
k.I_N \xrightarrow{*} & & k.I_N \xrightarrow{*} m_N
\end{array} \\
\begin{array}{ccc}
& & m_N = m'_N + k'.O_N \\
& & \xrightarrow{\sim} \\
& & k.I_N \xrightarrow{*} m'_N + k'.O_N
\end{array} \\
\downarrow \text{definition of } \sim \\
\begin{array}{ccc}
m_{NM} \xrightarrow{*} m'_{NM} & \xleftarrow{(1*)} & m_{NM} \\
\sim & & \sim \\
m'_N \xrightarrow{*} (k-k').O_N & & m'_N \xrightarrow{*} (k-k').O_N
\end{array} \\
\downarrow \text{definition of } \sim \\
m'_{NM} \xrightarrow{*} (k-k').O_{NM}
\end{array}$$

Figure 12: Structure of the proof of Theorem 4

This follows from further examination of the equality $m_N = m_{NM} + k'.O_N + [p^{k''}] - [p^{k'}] + k'.O_M - m_M^{k''}$. This time we look at the number of tokens in O_M . On the left-hand side there are clearly none. On the right hand side there are k' introduced by $k'.O_M$, and the only negative component $m_M^{k''}$ subtract no more than k'' of such tokens.

Now we continue with showing that $m'_N \sim m_{NM}$. This time from $m_{NM} + k'.O_{NM} = m_N - [p^{k''}] + m_M^{k''}$ we conclude $m_{NM} = m_N - [p^{k''}] + m_M^{k''} - k'.O_{NM}$ and again consider the two cases for $p \notin O_N$ or $p \in O_N$. If $p \notin O_N$, then $O_{NM} = O_N$ and so $m_{NM} = m_N - k'.O_N - [p^{k''}] + m_M^{k''} = m'_N - [p^{k''}] + m_M^{k''}$ and $m'_N(p) = m_N(p) - k'.O_N(p) = k'' - 0 = k''$ so $m'_N \sim m_{NM}$. If $p \in O_N$, then $O_{NM} = O_N - [p] + O_M$ and so $m_{NM} = m_N - [p^{k''}] + m_M^{k''} - k'.O_{NM} = m_N - [p^{k''}] + m_M^{k''} - k'.O_N + [p^{k'}] - k'.O_M = m_N - k'.O_N - [p^{k''-k'}] + m_M^{k''} - k'.O_M = m'_N - [p^{k''-k'}] + m_M^{k''} - k'.O_M$, so also then we can conclude that $m'_N \sim m_{NM}$ because $m_M^{k''} - k'.O_M \in S(M, k'' - k')$ and $k'' - k' = (m_N - k'.O_N)(p) = m'_N(p)$.

By the sub-soundness of N it then holds that $m'_N \xrightarrow{*} (k-k').O_N$. From (1*) it follows that $m_{NM} \xrightarrow{*} m'_{NM}$ such that $(k-k').O_N \sim m'_{NM}$, that is $m'_{NM} = (k-k').O_N - [p^x] + m_M^x$ with $m_M^x \in S(M, x)$ and $x = (k-k').O_N(p)$. If $p \notin O_N$, then $x = 0$ and $O_N = O_{NM}$, and therefore $m'_{NM} = (k-k').O_{NM}$. If $p \in O_N$, then $x = k-k'$ and therefore $m'_{NM} = (k-k').O_N - [p^{k-k'}] + m_M^{k-k'}$. Because M is sub-sound, it holds that $m_M^{k-k'} \xrightarrow{*} (k-k').O_M$, and since M is embedded in N and in this case $O_{NM} = O_N - [p] + O_M$, it follows that $m'_{NM} \xrightarrow{*} (k-k').O_N - [p^{k-k'}] + (k-k').O_M = (k-k').O_{NM}$. This way we have shown that in all cases $m_{NM} \xrightarrow{*} m'_{NM} \xrightarrow{*} (k-k').O_{NM}$ which concludes the proof.

$(k-k').O_{NM}$ which concludes the proof.

We now proceed with the case for place substitution in tWF nets. For that we will use the following lemma.

Lemma 9. *For every tWF net N with a place p and a disjoint pWF net M it holds that $\mathbf{pc}(N \otimes_p M) = \mathbf{pc}(N) \otimes_p M$.*

PROOF. Let $N = (P_N, T_N, F_N, I_N, O_N)$ with $p \in P_N$ and $M = (P_M, T_M, F_M, I_M, O_M)$. In both cases the same nodes are added, viz., those of M and p_i and p_o , see Figure 13. Clearly the edges F_M are added in the same way. Also in both cases afterwards $p_i \bullet = I_N$ and $\bullet p_o = O_N$ because N is a tWF net and $p \notin I_N$ and $p \notin O_N$. For nodes $p' \in I_M$ it holds in both cases that afterwards $\bullet p' = \bullet_N p$ if $p \notin I_N$ and $\bullet p' = \{p_i\}$ if otherwise. Similarly for nodes $p' \in O_M$ afterwards $p' \bullet = p \bullet_N$ if $p \notin O_N$ and $p' \bullet = \{p_o\}$. Finally, in both cases the final input set is $\{p_i\}$ and the final output set is $\{p_o\}$.

Theorem 10. *If a tWF net N is sub-sound and a disjoint pWF net M is sub-sound and p is a place in N then $N \otimes_p M$ is sub-sound.*

PROOF. Assume that a tWF net N is sub-sound and a pWF net M is sub-sound. By definition of sub-soundness for tWF nets it follows that $\mathbf{pc}(N)$ is sub-sound. By Theorem 8 it follows that $\mathbf{pc}(N) \otimes_p M$ is sub-sound. By Lemma 9 it then holds that $\mathbf{pc}(N \otimes_p M)$ is sub-sound. Finally, by definition of sub-soundness for tWF nets, it follows that $N \otimes_p M$ is sub-sound.

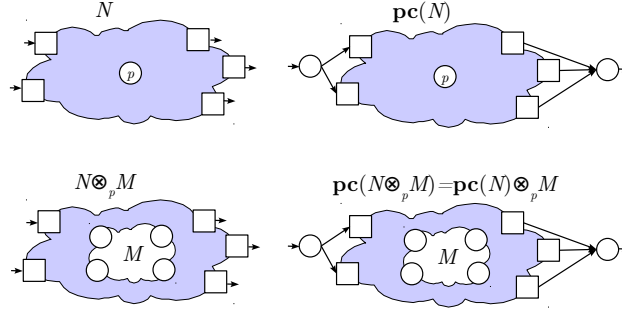


Figure 13: Place completion is semi-distributive in respect to place substitution

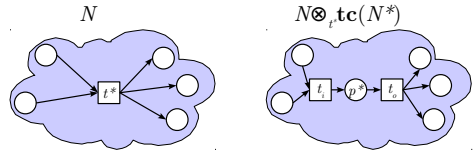


Figure 15: Sequential transition substitution

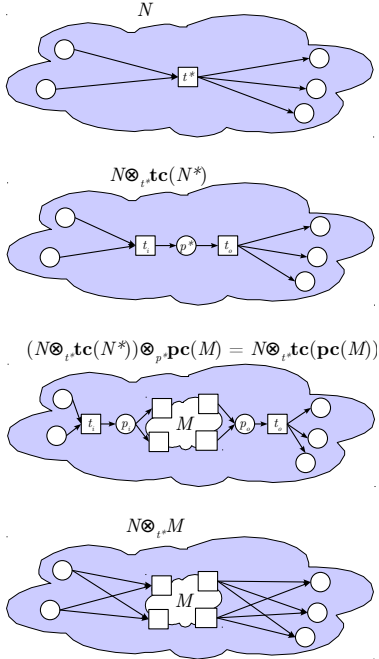


Figure 14: Transforming transition substitution to place substitution

We now proceed with showing that also transition substitution preserves sub-soundness. The proof strategy will be to show that this substitution is equivalent to a sequence of transformations with a place substitution as is illustrated in Figure 14. The top net is the original net N with transition t^* that is to be replaced with net M , the result of which, i.e., $N \otimes_{t^*} M$, is shown in the bottom. The sequence of transformations with a place substitution is shown in between. In the second row we see $N \otimes_{t^*} \mathbf{tc}(N^*)$ where N^* is a tWF consisting of transition completion of a single place p^* . As we show in Lemma 11, if N is sub-sound, then $N \otimes_{t^*} \mathbf{tc}(N^*)$ also is sub-sound. Next, we see the result of substituting the place p^* in $N \otimes_{t^*} \mathbf{tc}(N^*)$ with the pWF net $\mathbf{pc}(M)$. Finally, the input and output nodes introduced by the transition and place completions are removed, which also preserves sub-soundness as follows from Lemma 12 and Lemma 13.

We begin with the lemma that shows that, see Figure 15, if N is sub-sound, then $N \otimes_{t^*} \mathbf{tc}(N^*)$ also is sub-sound.

Lemma 11. *If N is a pWF net with a transition t^* and N^* a pWF net that consists of only a single place p^* , then $N \otimes_{t^*} \mathbf{tc}(N^*)$ is sub-sound if N is sub-sound.*

PROOF. Let $M = N \otimes_{t^*} \mathbf{tc}(N^*)$. We define a relation $\sim \subseteq \mathbf{M}_N \times \mathbf{M}_M$ such that $m_N \sim m_M$ represents the fact that m_N is the same as m_M except that all (say k) tokens are removed from p^* and k tokens are added to each of $t_o \bullet_M$, or in other words, t_o is fired k times, where t_o is the

output transition added in $\mathbf{tc}(N^*)$. More formally: $m_N \sim m_M$ iff $m_N = m_M - [p^*]^k + k.(t_o \bullet_M)$ where $k = m_M(p^*)$.

It can then be shown that \sim indeed defines a kind of bisimilarity, i.e., it holds that (A*) if $m_N \xrightarrow{\sigma}_N m'_N$ and $m_N \sim m_M$, then $m_M \xrightarrow{\sigma}_M m'_M$, and (B*) if $m \xrightarrow{\sigma}_M m'_M$, then there is a marking m_N such that $m \xrightarrow{\sigma}_N m_N$ and $m_N \sim m_M$. Informally this can be shown with induction on the length of σ . For each transition s in σ we then distinguish for (A*) the cases where $s = t^*$ or not. Likewise for (B*) we distinguish the cases where $s = t_i$ or s is a transition in M not equal to t_i nor t_o (we will argue that with these assumptions $s \neq t_o$).

We now show by using (A*) and (B*) that M is sub-sound if N is sub-sound. Note that, by construction of M , N and M have the same input set I and output set O . Assume that N is sub-sound and that $k.I \xrightarrow{*}_M (m_M + k'.O)$. By (B*) it follows that $k.I \xrightarrow{*}_N m_N$ such that $m_N \sim (m_M + k'.O)$ that is $m_N = m_M + k'.O - [p^{*k'}] + k'.(t_o \bullet_M)$ where $k'' = (m_M + k'.O)(p^*)$. Since $p^* \notin O$ we can assume that $k'' = m_M(p^*)$ and that $m_N = (m_M - [p^{*k''}] + k''.(t_o \bullet_M)) + k'.O$, i.e., $k'.O \leq m_N$ and get $(m_N - k'.O) \sim m_M$. From the sub-soundness of N it follows that $(m_N - k'.O) \xrightarrow{*}_N (k - k').O$. Finally, by (A*) it follows that $m_M \xrightarrow{*}_M (k - k').O$ which completes the proof of sub-soundness of M .

We will now formally show the missing (A*) and (B*). We start with the following facts:

(A) If $m_N \xrightarrow{t}_N m'_N$ and $m_N \sim m_M$, then $m_M \xrightarrow{*}_M m'_M$. **Proof:** If $m_M(p^*) = k$, then we can fire k times t_o and so $m_M \xrightarrow{*}_M m'_M = m_M - [p^{*k}] + k.(t_o \bullet_M)$. Since $m_N \sim m_M$ we also have that $m_N = m_M - [p^{*k}] + k.(t_o \bullet_M)$ and so $m'_M = m_N$, i.e., $m_M \xrightarrow{*}_M m'_M$. Either (i) $t \neq t^*$ or (ii) $t = t^*$. If (i), then by construction of M we have $\bullet_N t = \bullet_M t$ and $t \bullet_N = t \bullet_M$ and so from $m_N \xrightarrow{t}_N m'_N$ it follows $m_N \xrightarrow{t}_M m'_N$. Thus we have shown that $m_M \xrightarrow{*}_M m_N \xrightarrow{t}_M m'_N$. If (ii), then by construction of M we have $\bullet_N t = \bullet_M t_i$ and $t \bullet_N = t_o \bullet_M$ and so from $m_N \xrightarrow{t}_N m'_N$ and the fact that $t_i \bullet_M = \bullet_M t_o$ it follows $m_N \xrightarrow{t_i, t_o}_M m'_N$. Thus we have shown that $m_M \xrightarrow{*}_M m_N \xrightarrow{t_i, t_o}_M m'_N$.

(B) If $m \xrightarrow{t}_M m_M$, then there is an m_N such that $m \xrightarrow{*}_N m_N$ and $m_N \sim m_M$. **Proof:** Because we assumed that m is also a marking of N it holds that $m(p^*) = 0$, so either (i) $t \notin \{t_i, t_o\}$ or (ii) $t = t_i$. If (i), then by construction of M we have $\bullet_N t = \bullet_M t$ and $t \bullet_N = t \bullet_M$ and so from $m \xrightarrow{t}_M m_M$ it follows $m \xrightarrow{t}_N m_M$. Of course $m_M \sim m_M$. If (ii), then by

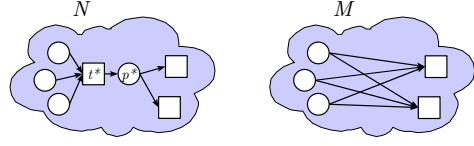


Figure 16: Transition-place pair removal

construction of M we have $\bullet_N t^* = \bullet_M t_i$ and so from $m \xrightarrow{t_i}_M m_M$ it follows $m \xrightarrow{t^*}_N m_N$ for some m_N . We have $m_N = m - \bullet_N t^* + t^* \bullet_N = m - \bullet_M t_i + t_o \bullet_M$. On the other hand $m_M = m - \bullet_M t_i + t_i \bullet_M = m - \bullet_M t_i + p^*$. By combining these two we get $m_N = m_M - p^* + t_o \bullet_M$ and because $m(p^*) = 0$ we have $m_M(p^*) = 1$, so by definition $m_N \sim m_M$.

The facts (A) and (B) can be generalized by induction on the length of σ as follows: (A*) if $m_N \xrightarrow{\sigma}_N m'_N$ and $m_N \sim m_M$, then $m_M \xrightarrow{\sigma}_M m'_M$, and (B*) if $m \xrightarrow{\sigma}_M m'_M$, then there is a marking m_N such that $m \xrightarrow{\sigma}_N m_N$ and $m_N \sim m_M$.

We now proceed with lemmas that show that the removal of t_i and p_i as well as p_o and t_o preserves sub-soundness. These results are similar to those of the abstraction rule of [6].

Lemma 12. Let N be a pWF net with transition t^* and place p^* such that $t^* \bullet_N = p^*$, $\bullet_N p^* = t^*$ and p^* is not an input nor output place and there are no edges between $\bullet_N t^*$ and $p^* \bullet_N$. Furthermore, let M be the pWF net that is obtained from N if we remove t^* and p^* and add all the edges in $\bullet_N t^* \times p^* \bullet_N$ as illustrated in Figure 16. Then M is sub-sound if N is sub-sound.

PROOF. We define a similarity relation $\sim \subseteq \mathbf{M}_N \times \mathbf{M}_M$ such that $m_N \sim m_M$ represents the fact that m_M is the same as m_N except that all (say k) tokens are removed from p^* and k tokens are added to each of $\bullet_N t^*$, or in other words, t^* is fired k times in reverse. More formally: $m_N \sim m_M$ holds iff $m_M = m_N - [p^{*k}] + k.(\bullet_N t^*)$ where $k = m_N(p^*)$.

It can then be shown that \sim indeed defines a kind of bisimilarity, i.e., it holds that (D*) if $m_N \xrightarrow{\sigma}_N m'_N$ and $m_N \sim m_M$, then there is a marking m'_M such that $m_M \xrightarrow{\sigma}_M m'_M$ and $m'_N \sim m'_M$, (E*) if $m_M \xrightarrow{\sigma}_M m'_M$ and $m_N \sim m_M$, then there is a marking m'_N such that $m_N \xrightarrow{\sigma}_N m'_N$ and $m'_N \sim m'_M$, and (F*) if $m \xrightarrow{\sigma}_M m_M$, $m \xrightarrow{\sigma}_N m_N$, $m_N \sim m_M$ and $m_N(p^*) > 0$ then from σ we can construct σ' by removing the last $m_N(p^*)$ occurrences of t^* and get $m \xrightarrow{\sigma'}_N m'_N$, $m'_N \sim m_M$ and

$m'_N(p^*) = 0$. Informally this can be shown with induction on the length of σ where for each transition t in σ we distinguish for (D*) the cases where $t = t^*$ and if not then $p^* \in \bullet_N t$ or not. Likewise, for (E*) we distinguish the cases where $p^* \in \bullet_N t$ or not. Finally, for (F*) we observe that all $m_N(p^*)$ tokens in p^* had to be placed there by t^* during σ and that the last of those tokens is not needed by the following transitions of σ .

Now, using (D*), (E*) and (F*), we show that M is sub-sound if N is sub-sound. Note that, by construction, N and M have the same input set I and output set O , and that $O \sim O$.

Assume that $k.I \xrightarrow{*}_M (m_M + k'.O)$. By (E*) it follows that $k.I \xrightarrow{*}_N m_N$ such that $m_N \sim (m_M + k'.O)$. By (F*) $k.I \xrightarrow{*}_N m'_N$ where $m'(p^*) = 0$ and $m'_N \sim (m_M + k'.O)$. By definition of \sim the last two give $m'_N = (m_M + k'.O)$, i.e., $k.I \xrightarrow{*}_N (m_M + k'.O)$. Now by sub-soundness of N it follows that $m_M \xrightarrow{*}_N (k - k').O$. By (D*) $m_M \xrightarrow{*}_M m'_M$ and $(k - k').O \sim m'_M$. Since $(k - k').O(p^*) = 0$ we get $m'_M = (k - k').O$.

We now will formally show the missing (D*), (E*) and (F*). We start with the following facts:

(A) If $m_N \sim m_M$ and t such that $t \neq t^*$ and $\bullet_N t \leq m_N$ then $\bullet_M t \leq m_M$. **Proof:** Assume that $m_N \sim m_M$, $t \neq t^*$ and $\bullet_N t \leq m_N$. From $m_N \sim m_M$ it follows that $m_M = m_N - [p^{*k}] + k.(\bullet_N t^*)$ where $k = m_N(p^*)$. Consider the case where $p^* \in \bullet_N t$. Then $\bullet_M t = \bullet_N t - [p^*] + \bullet_N t^* \leq m_N - [p^{*k}] + k.(\bullet_N t^*) = m_M$ where the first equality follows from the definition of M and the inequality from the observation that in this case $k \geq 1$. Consider the other case where $p^* \notin \bullet_N t$. Here from $\bullet_N t \leq m_N$ it follows that $\bullet_N t \leq m_N - [p^{*k}] + k.(\bullet_N t^*)$ and we get $\bullet_M t = \bullet_N t \leq m_N - [p^{*k}] + k.(\bullet_N t^*) = m_M$.

(B) If $\bullet_M t \leq m$ then there is an $m_N \sim m$ such that $m \xrightarrow{*}_N m_N$, $\bullet_N t \leq m_N$. **Proof:** Assume that $\bullet_M t \leq m$. Consider the case where $p^* \notin \bullet_N t$. Then $\bullet_N t = \bullet_M t \leq m$ and so we can take $m_N = m$. Consider the other case where $p^* \in \bullet_N t$. By the construction $\bullet_N t^* \leq \bullet_M t \leq m$, i.e., t has to be enabled in N . Let m_N be a marking such that $m \xrightarrow{t^*}_N m_N$ that is $m_N = m - \bullet_N t^* + p^*$. Since $\bullet_N t = \bullet_M t - \bullet_N t^* + p^*$ this implies that $\bullet_N t \leq m_N$.

(C) If $m_N \sim m_M$ and $m_N \xrightarrow{t}_N m'_N$ and $m_M \xrightarrow{t}_M m'_M$ then $m'_N \sim m'_M$. **Proof:** Assume that $m_N \sim m_M$ and $m_N \xrightarrow{t}_N m'_N$ and $m_M \xrightarrow{t}_M m'_M$. Because $m_N \sim m_M$, $m_M = m_N - [p^{*k}] + k.(\bullet_N t^*)$ where $k = m_N(p^*)$. Because $m_N \xrightarrow{t}_N m'_N$, $m'_N = m_N - \bullet_N t + t\bullet_N$. Because $m_M \xrightarrow{t}_M m'_M$, $m'_M = m_M - \bullet_M t + t\bullet_M$. By construction and because $t \neq t^*$, $t\bullet_M = t\bullet_N$. Now either (i) $p^* \in \bullet_N t$ or (ii) $p^* \notin \bullet_N t$. If (i) then $k \geq 1$ and $\bullet_M t = \bullet_N t - [p^*] + \bullet_N t^*$. It follows that $m'_M = (m_N - [p^{*k}] + k.(\bullet_N t^*) - (\bullet_N t -$

$[p^*] + \bullet_N t^*) + t\bullet_N) = m_N - [p^{*(k-1)}] + (k - 1).(\bullet_N t^* - \bullet_N t + t\bullet_N) = m_N - \bullet_N t + t\bullet_N - [p^{*(k-1)}] + (k - 1).(\bullet_N t^* = m'_N - [p^{*(k-1)}] + (k - 1).(\bullet_N t^* with $m'_N(p^*) = k - 1$ since $p^* \in \bullet_N t$ and $p^* \notin t\bullet_N$. Thus $m'_N \sim m'_M$. Consider the other case (ii) where $p^* \notin \bullet_N t$. Then $\bullet_M t = \bullet_N t$ and therefore $m'_M = (m_N - [p^{*k}] + k.(\bullet_N t^*) - \bullet_N t + t\bullet_N = m_N - \bullet_N t + t\bullet_N - [p^{*k}] + k.(\bullet_N t^*) = m'_N - [p^{*k}] + k.(\bullet_N t^*) with $m'_N(p^*) = k$, since $p^* \notin \bullet_N t$ and $p^* \notin t\bullet_N$. Thus $m'_N \sim m'_M$.$$

(D) If $m_N \xrightarrow{t}_N m'_N$ and $m_N \sim m_M$ then there is a marking m'_M such that $m_M \xrightarrow{*}_M m'_M$ and $m'_N \sim m'_M$.

Proof: Assume that $m_N \xrightarrow{t}_N m'_N$ and $m_N \sim m_M$. Now either (i) $t = t^*$ or (ii) $t \neq t^*$. If (i) then $m'_N \sim m_M$ and so we can take $m'_M = m_M$. Consider the case (ii) where $t \neq t^*$. By (A) it then holds that t is enabled in m_M for M , and so $m_M \xrightarrow{t}_M m'_M$ for some m'_M . By (C) it then follows that $m'_N \sim m'_M$.

(E) If $m \xrightarrow{t}_M m_M$ then there is a marking m_N such that $m \xrightarrow{*}_N m_N$ and $m_N \sim m_M$. **Proof:** Assume that $m \xrightarrow{t}_M m_M$. By (B) there is an m_N such that $m \xrightarrow{*}_N m_N$, $\bullet_N t \leq m_N$ and $m_N \sim m$. Since $\bullet_N t \leq m_N$ it holds that $m_N \xrightarrow{t}_N m'_N$ for some m'_N . By (C) it then follows that $m'_N \sim m$.

(F) If $m \xrightarrow{*}_M m_M$, $m \xrightarrow{\sigma}_N m_N$, $m_N \sim m_M$ and $m_N(p^*) > 0$ then from σ we can construct σ' by removing the last occurrence of t^* and get $m \xrightarrow{\sigma'}_N m'_N$, $m'_N \sim m_M$ and $m'_N(p^*) = m_N(p^*) - 1$. **Proof:** Assume that $m \xrightarrow{*}_M m_M$, $m \xrightarrow{\sigma}_N m_N$, $m_N \sim m_M$ and $m_N(p^*) > 0$. Since m is a marking of both N and M , it does not place any tokens in p^* which is not present in M . So all $m_N(p^*)$ tokens in p^* had to be placed there during σ by firing t^* , which is the only transition that can do that, and the token placed there as last is not needed by the following transitions of σ . This is due to the fact that we do not distinguish individual tokens of a place and without the loss of generality we can assume that places act as FIFO queues for tokens. Thus a valid firing sequence σ' can be constructed from σ by removing the last occurrence of t^* . Let $m \xrightarrow{\sigma'}_N m'_N$. By the definition of σ' $m'_N = m_N - p^* + \bullet_N t^*$. It follows that $m'_N \sim m_M$ and $m'_N(p^*) = m_N(p^*) - 1$.

With induction on the length of σ it follows that (D*) if $m_N \xrightarrow{\sigma}_N m'_N$ and $m_N \sim m_M$ then there is a marking m'_M such that $m_M \xrightarrow{*}_M m'_M$ and $m'_N \sim m'_M$, (E*) if $m_M \xrightarrow{\sigma}_M m'_M$ and $m_N \sim m_M$ then there is a marking m'_N such that $m_N \xrightarrow{*}_N m'_N$ and $m'_N \sim m'_M$, and (F*) if $m \xrightarrow{*}_M m_M$, $m \xrightarrow{\sigma}_N m_N$, $m_N \sim m_M$ and $m_N(p^*) > 0$

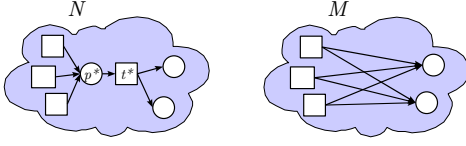


Figure 17: Place-transition pair removal

then from σ we can construct σ' by removing the last $m_N(p^*)$ occurrences of t^* and get $m \xrightarrow{\sigma'} m'_N, m'_N \sim m_M$ and $m'_N(p^*) = 0$.

Lemma 13. *Let N be a pWF net with place p^* and transition t^* such that $p^* \bullet_N = t^*$, $\bullet_N t^* = p^*$ and p^* is not an input nor output place and there are no edges between $\bullet_N p^*$ and $t^* \bullet_N$. Furthermore, let M be the pWF net that is obtained from N if we remove p^* and t^* and add all the edges in $\bullet_N p^* \times t^* \bullet_N$ as illustrated in Figure 17. Then M is sub-sound if N is sub-sound.*

PROOF. The proof proceeds analogously to that of the preceding Lemma 12 with the relation $\sim \subseteq \mathbf{M}_N \times \mathbf{M}_M$ redefined such that $m \sim m'$ iff $m' = m - [p^{*k}] + k.(t^* \bullet_N)$ where $k = m(p^*)$.

We are now ready to prove that sub-soundness is preserved by transition substitution.

Theorem 14. *If a pWF net N is sub-sound and a disjoint tWF net M is sub-sound and t^* is a transition in N , then $N \otimes_{t^*} M$ is sub-sound.*

PROOF. Let $N = (P, T, F, I, O)$ be a sub-sound pWF net containing a transition t , and $M = (P', T', F', I', O')$ a sub-sound tWF net. Furthermore, let N^* be a pWF net consisting of a single new place $p^* \notin P \cup P'$. We will construct $N \otimes_{t^*} M$ by a sequence of transformations and substitutions where the sub-soundness of the result of each step will follow from the sub-soundness of the nets used as components.

Consider the sequence of transformation in Figure 14. In the top we start with N which by assumption is sub-sound. We first substitute t^* with $\mathbf{tc}(N^*)$ and get $N \otimes_{t^*} \mathbf{tc}(N^*)$ which by Lemma 11 is sub-sound if N is sub-sound. Then we substitute p^* with $\mathbf{pc}(M)$ and get $(N \otimes_{t^*} \mathbf{tc}(N^*)) \otimes_{p^*} \mathbf{pc}(M)$. Here the sub-soundness follows from Theorem 8 and the fact that a place completion a sub-sound tWF net is sub-sound by definition. Finally we remove nodes t_i and p_i as well as p_o and t_o by applying Lemmas 12 and 13 respectively. This concludes the proof that the resulting net $N \otimes_{t^*} M$ is sub-sound.

Theorem 15. *If a tWF net N is sub-sound and a disjoint tWF net M is sub-sound and t is a transition in N then $N \otimes_t M$ is sub-sound.*

PROOF. Assume that N is sub-sound tWF net with a transition t and M a sub-sound tWF net. By Theorem 14 it follows that $\mathbf{pc}(N) \otimes_t M$ is sub-sound. Since by Lemma 9 it holds that $\mathbf{pc}(N \otimes_t M) = \mathbf{pc}(N) \otimes_t M$, it follows that $\mathbf{pc}(N \otimes_t M)$ is sub-sound. By definition of sub-soundness of tWF nets it then holds that $N \otimes_t M$ is sub-sound.

Corollary 16. *If N and M are disjoint sub-sound WF nets and n is a node in N then $N \otimes_n M$ (if defined) is a sub-sound WF net.*

PROOF. This follows from the fact that Theorem 8, Theorem 10, Theorem 14 and Theorem 15 cover all possible combinations of N and M being pWF nets or tWF nets.

5. Sub-soundness of AND-OR nets

In this section we show that all AND-OR nets are sub-sound. First we show that the AND nets and OR nets from which AND-OR nets are generated are.

Theorem 17. *Every one-input one-output pOR net is sub-sound.*

PROOF. Since in OR nets transitions cannot have multiple input/output places it can be shown by induction on the length of σ that (A) if $|m| = k$ and $m \xrightarrow{\sigma} m'$ then $|m'| = k$. Let $I_N = \{p_i\}$ and $O_N = \{p_o\}$. For each place p in a pOR net N it holds that $[p_i] \xrightarrow{*} [p]$ and $[p] \xrightarrow{*} [p_o]$ since there must be paths from p_i to p and from p to p_o and each transition in those paths has one input edge and one output edge. Thus, it also follows that (B) if $|m| = k$, then $k.[p_i] \xrightarrow{*} m$ and $m \xrightarrow{*} k.[p_o]$.

We now show the sub-soundness requirement. Assume that $k.I_N \xrightarrow{*} (m + k'.O_N)$. Since $|k.I_N| = |k.[p_i]| = k$ it follows by (A) that $|m + k'.O_N| = k$. Since $|m + k'.O_N| = |m| + |k'.O_N|$ and $|k.O_N| = |k.[p_o]| = k * |[p_o]| = k$ it follows that $|m| = k - k'$. By (B) it then follows that $m \xrightarrow{*} (k - k').[p_o] = (k - k').O_N$.

Theorem 18. *Every tOR net is sub-sound.*

PROOF. Consider a tOR net N . By the definition of *-soundness of tWF nets it holds that N is *-sound if $\mathbf{pc}(N)$ is *-sound. Observe that $\mathbf{pc}(N)$ is an one-input one-output pAND net, because N by definition it does not have and incoming edges of the input places nor outgoing edges of the output places. By Theorem 17 it holds that $\mathbf{pc}(N)$ is sub-sound and therefore *-sound. By Lemma 7 it follows that N is sub-sound.

Theorem 19. *Every pAND net is sub-sound.*

PROOF. Consider a pAND net N . Thanks to the limit on the number of incoming edges of the input places and outgoing edges of the output places in the definition of AND net $\mathbf{tc}(N)$ is a one-input one-output tAND net. Also $\mathbf{tc}(N)$ does not have incoming edges of the input transition nor outgoing edges of the output transition. By Theorem 17 in [14] we get that $\mathbf{pc}(\mathbf{tc}(N))$ is $*$ -sound which by definition gives proves that $\mathbf{tc}(N)$ is $*$ -sound. By Theorem 1 it follows that N is sub-sound and therefore $*$ -sound. Therefore by Lemma 6 it follows that N is sub-sound.

Theorem 20. *Every one-input one-output tAND net is sub-sound.*

PROOF. Consider a one-input, one-output tAND net N . It's input transition cannot have incoming edges nor its output transition cannot have outgoing edges, since those would have to introduce cycles. By Theorem 19 it follows that $\mathbf{pc}(N)$ is sub-sound, so also $*$ -sound, and thus N is $*$ -sound. Therefore by Lemma 7 it follows that N is sub-sound.

Corollary 21. *All AND-OR nets are sub-sound.*

PROOF. By the Theorem 17, Theorem 18, Theorem 20 and Theorem 19 the initial nets are all sub-sound, and by Corollary 16 substitution preserves sub-soundness.

6. Future Research

The class of AND-OR nets can be researched further in several ways. One direction could be to attempt to characterize the class in terms of syntactic and semantic properties. As was shown all the nets in it are sound, even sub-sound, and it is also not hard to see that they are all free-choice nets, but it certainly not true that the class contains all sub-sound free-choice nets as is show in Theorem 22. So it remains open which semantic property characterizes the AND-OR nets.

Theorem 22. *Not all free-choice sub-sound workflow nets are AND-OR nets.*

PROOF. The counterexample is given in Figure 18 (a) (taken from [14]).

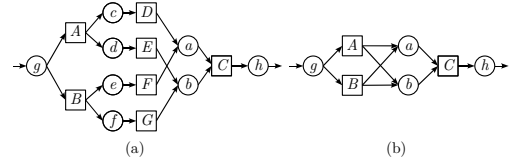


Figure 18: Counterexample for the completeness of AND-OR nets

Another potential research direction is the extension of the class by introducing new forms of substitution that still can be considered hierarchical. For example, it might be allowed that not only substitute nodes but also edges: an edge from a place to a transition could be replaced with a workflow net starting with a single place and ending with a single transition. In general such substitutions do not preserve sub-soundness, but they can be syntactically restricted such that they do. To illustrate, such substitutions could be used to generate Figure 18 (a) from the AND-OR net in Figure 18 (b) by substituting the edges (A, a) , (A, b) , (B, a) and (B, b) .

Yet another possible generalization can be achieved by weakening the requirement that a substitution links all the input and output nodes in the same way. For example, it could be allowed that a transition is replaced with a tAND net with a single input transition and several output transitions such that (1) each output transition in the tAND net is linked to at least one place in the postset of the replaced transition and (2) each place in the postset of the replaced transition is linked with exactly one output transition in the tAND net. Also this would allow us to generate Figure 18 (a) from the AND-OR net in Figure 18 (b) by substituting the transitions A and B .

7. Conclusions

We have presented an approach for designing sound workflow nets in a structured way. This method is based on the notion of a substitution of one node by a workflow net with input and output nodes being of the same type as the substituted node. The substituted nets can have multiple inputs and outputs, which is an extension to the previously considered substitutions as it allows to generate more general class of nets. We have identified a notion of soundness that is preserved by such substitutions and corrected a small omission in an earlier similar method.

References

- [1] W. Reisig, C. Adam Petri, Petri net, Scholarpedia 3 (2008) 6477.

- [2] W. M. P. van der Aalst, The application of Petri nets to workflow management., *Journal of Circuits, Systems, and Computers* 8 (1998) 21–66.
- [3] W. M. P. van der Aalst, K. M. van Hee, A. H. M. ter Hofstede, N. Sidorova, H. M. W. Verbeek, M. Voorhoeve, M. T. Wynn, Soundness of workflow nets: classification, decidability, and analysis., *Formal Asp. Comput.* 23 (2011) 333–363.
- [4] B. A., R. Ben Ayed, S. Z., Workflow soundness verification based on structure theory of Petri nets, *International Journal of Computing & Information Sciences* 5 (2007) 51–61.
- [5] J. Esparza, M. Silva, On the analysis and synthesis of free choice systems., in: G. Rozenberg (Ed.), *Applications and Theory of Petri Nets*, volume 483 of *Lecture Notes in Computer Science*, Springer, 1989, pp. 243–286.
- [6] J. Desel, J. Esparza, *Free Choice Petri Nets* (Cambridge Tracts in Theoretical Computer Science), Cambridge University Press, New York, NY, USA, 2005.
- [7] M. Dumas, M. L. Rosa, J. Mendling, R. Mäesalu, H. A. Reijers, N. Semenenko, Understanding business process models: The costs and benefits of structuredness, in: J. Ralyté, X. Franch, S. Brinkkemper, S. Wrycza (Eds.), *Advanced Information Systems Engineering - 24th International Conference, CAiSE 2012*, Gdansk, Poland, June 25-29, 2012. *Proceedings*, volume 7328 of *Lecture Notes in Computer Science*, Springer, 2012, pp. 31–46.
- [8] A. Polyvyanyy, L. García-Bañuelos, M. Dumas, Structuring acyclic process models, *Information Systems* 37 (2012) 518 – 538.
- [9] W. Brauer, R. Gold, W. Vogler, A survey of behaviour and equivalence preserving refinements of Petri nets, *Advances in Petri Nets* 1990 (1991) 1–46.
- [10] I. Suzuki, T. Murata, A method for stepwise refinement and abstraction of Petri nets, *Journal of Computer and System Sciences* 27 (1983) 51–76.
- [11] Z. Peng, Q. Mei, L. Xikui, The behavior properties of refinement of Petri nets and application in parallel programming, *Computer Science and Information Engineering, World Congress on* 3 (2009) 330–333.
- [12] J. Padberg, M. Gajewsky, C. Ermel, Rule-based refinement of high-level nets preserving safety properties, *Science of Computer Programming* 40 (2001) 97–118.
- [13] H. Huang, T. Y. Cheung, W. M. Mak, Structure and behavior preservation by Petri-net-based refinements in system design, *Theoretical Computer Science* 328 (2004) 245–269.
- [14] K. M. van Hee, N. Sidorova, M. Voorhoeve, Soundness and separability of workflow nets in the stepwise refinement approach, in: W. M. P. van der Aalst, E. Best (Eds.), *ICATPN*, volume 2679 of *Lecture Notes in Computer Science*, Springer, 2003, pp. 337–356.
- [15] P. Chrzastowski-Wachtel, B. Benatallah, R. Hamadi, M. O’Dell, A. Susanto, A top-down Petri net-based approach for dynamic workflow modeling, in: *Proceedings of BPM’03*, Springer-Verlag, Berlin, Heidelberg, 2003, pp. 336–353.
- [16] K. van Hee, J. Hidders, G.-J. Houben, J. Paredaens, P. Thiran, On the relationship between workflow models and document types, *Inf. Syst.* 34 (2009) 178–208.
- [17] J. Sroka, P. Chrzastowski-Wachtel, J. Hidders, On generating *-sound nets with substitution, *Application of Concurrency to System Design, International Conference on* 0 (2011) 3–12.